Scale Aware Spatially Guided Mapping

Parul (2016csb1048)
Prerna Garg (2016csb1050)

Table Of Contents

References

1) https://ieeexplore.ieee.org/abstract/document/7436662/
2) https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13003
3) https://ieeexplore.ieee.org/abstract/document/6319316/
4) https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01645.x

Overview

Various Image Stylization techniques such as **Cartoonization** and **Pencil Drawing** are based upon filtering techniques where we separate the textures from the dominant edges using the difference in their gradient magnitudes.

However, scale is also an important factor that should be considered in such processes.

Image contents at large scales often exhibit the spatial configuration of the scene, while visual patterns at small scales are usually textures or even noises.

Here, we implement a Scale-Aware Spacially Guided Filter that inculcates both scale as well as magnitude aspects.

Our Basic Initiative for project

1. Proposal and initialisation of a scale-aware spatially guided mapping (SaSGM) model, which specifies the influence of image structures in terms of both edge magnitude and scale.

2. Analysis of rolling guidance filter used in the first step of scale aware mapping in terms of functionality, order and time taken for computation.

3. Proposal, implementation and interpretation of a fast guided rolling filter through the use of probability theorems with speed improvement of around 150-200 times.

4. Introduce this model into applications such as detail enhancement and image stylization including pencil sketch and cartoonization.

The SaSGM Model

- Specify a set of scales $\{ \sigma_i \}$ which would represent the size of an edge.
- Create a multi channel like pipeline where each channel represents the observation $I_{\sigma i}$ i.e. of input image under a scale σ_i, therefore $I_{\sigma i}$ only contains structures that are smaller than σ_i using rolling guidance filter
- Apply gradient convolution to the set $\{ I_{\sigma is} \}$ to generate their edge responses $\nabla I_{\sigma is}$ which would henceforth make up a level-of-detail measure of an image edge response.
- Measure the spatial influence of each $\nabla I_{\sigma i}$ under its scale σ_i using a spatial indicator function $f_{\sigma i}(j)$ at (x,y) defined as:

The SaSGM Model (continued)

$$f_{\sigma i(x,y)}(j) = \begin{cases} 1 & \text{if } |\nabla \mathrm{I}_{\sigma i}(x,y)| = j \\ 0 & \text{otherwise} \end{cases},$$

- Here j is the normalized intensity in the edge response. For $j \in \{1,\ldots,n_b\}$ the spatial influence can be approximated as:

$$\mathbf{M}^{a}_{\sigma i} = \sum_{n_b} (\mathbf{f}_{\sigma i}(j) * G(\sigma_i))w(j),$$

where $*$ $G(\sigma_i)$ means convolution with a 2D Gaussian mask of radius σ_i and $w(j)$ is the weight of the j^{th} intensity bin.By countering the histogram h_j of n_b bins, the weights can be easily approximated as $1-h_j$, where large values are generated for bins of strong edges and vice versa.

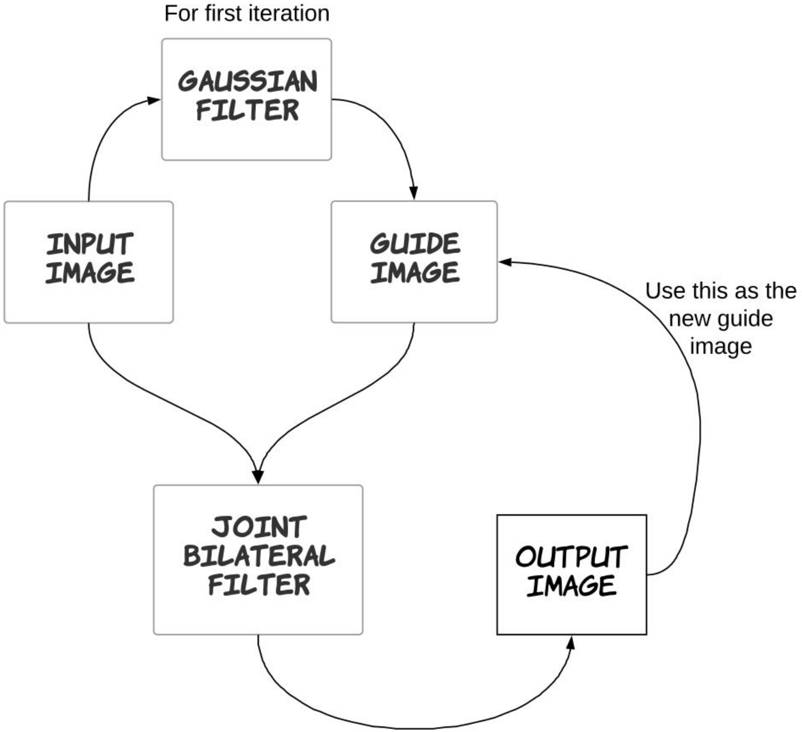
The SaSGM Model

- Next, we simply combine multiple $\mathbf{M}_{\sigma i}^{a}$ maps together to generate the whole accumulated map M^a :

$$\mathbf{M}^{a} = \sum_{i} \mathbf{M}_{\sigma i}^{a}.$$

- To further refine M^a, we again apply a joint bilateral filter by taking the original image I as the guidance for reconstructing the image patches. The guided image filter assumes a linear dependency between the patches from the output M and guidance image I, so M has a more faithful morphology than M^a of the original image I.

RGF (Rolling Guidance Filter)



OBSERVATION:
The variance of noise is greater than variance of an edge.
The gaussian filter effectively removes noise but also blurs the edges which is undesirable.
AIM:
Suppress noise without blurring the actual edges.
SOLUTION:
Apply gaussian filter to remove noise, retrieve the sharpness of the actual edges.

Joint Bilateral Filter (JBF)

1) Joint bilateral filter has been designed to solve the purpose of achieving removal of noise without blurring the edges where I is the gaussian output for input image for the first iteration and J for the subsequent iterations.

$$J(p) = \frac{1}{K_p} \sum_{q \in R(p)} \exp\left(-\frac{|p-q|^2}{2\sigma_s^2}\right) \cdot \exp\left(-\frac{|G(p)-G(q)|^2}{2\sigma_r^2}\right) \cdot I(q)$$

2) For every pixel p we take a neighbourhood Ω_q and two contributions spatial and color i.e. f (|| p - q ||) and g(|| G_q - G_p ||) where they are defined as follows:

g(|| G_q - G_p ||) = $\exp\left(-\frac{|G(p)-G(q)|^2}{2\sigma_r^2}\right)$

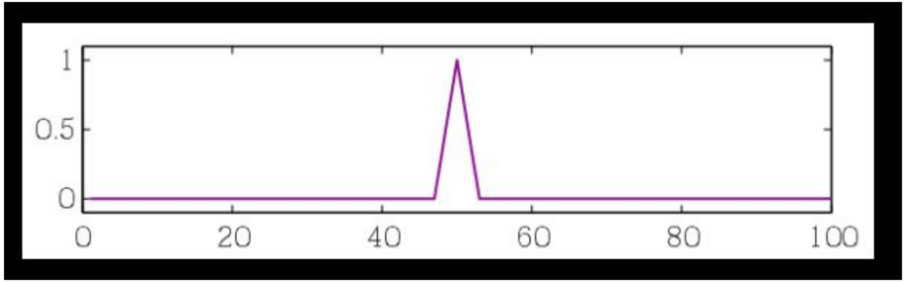
and f (|| p - q ||) = $\exp\left(-\frac{|p-q|^2}{2\sigma_s^2}\right)$

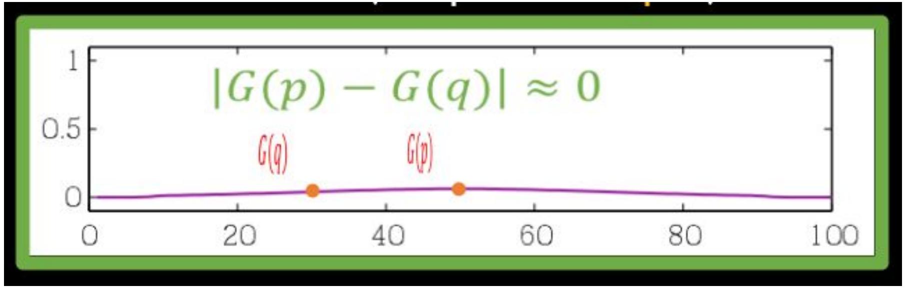
Joint Bilateral Filter (JBF continued)

1) Lesser the spatial distance between p and q more is its contribution in the output intensity of image and lesser the difference between color intensities of p and q more is the contribution of q in the output image.
2) When p and q are closer f (|| q - p ||) will be higher, similarly when I_q and I_p are closer the value of g (|| I_q - I_p ||) will be higher.
3) For pixel representing noise I_q and I_p i.e. outputs of gaussian filter on input will be very close values and hence the term in the exponent will tend to 0 and g (|| I_q - I_p ||) will become 1 and the equation would represent a normal gaussian filtering for p.
4) But for pixel representing a true edge I_q and I_p i.e. outputs of gaussian filter on input will be significantly different and hence the exponential term would be <1 giving appropriate fraction to q's contribution thus preserving p's intensity variation as in the input image.
5) This phenomenon has been illustrated in the subsequent slides.

Concept of Guide Image

Small Structures (Noise)



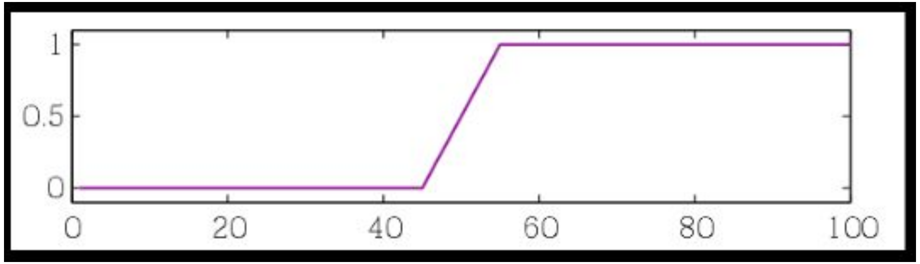


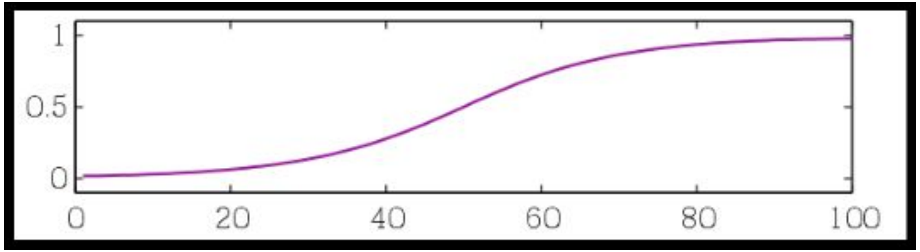
$$J(p) = \frac{1}{K_p} \sum_{q \in R(p)} \exp\left(-\frac{|p-q|^2}{2\sigma_s^2}\right) \cdot \overset{\approx 1}{\boxed{\exp\left(-\frac{|G(p)-G(q)|^2}{2\sigma_r^2}\right)}} \cdot I(q)$$

For noise the intensity will be a sudden peak with respect to the neighbourhood. The gaussian filter will suppress the peak as shown in the second image. Therefore values of G(p) and G(q) will be approximately equal. Hence the contribution of guide image to the equation of joint bilateral filter will approximate to 1 rendering it as the equation of gaussian filter for this particular pixel representing noise. Therefore , When we apply joint bilateral filter to the image the noise would be successfully suppressed.

Concept of Guide Image (continued)

Large Structures (Noise)

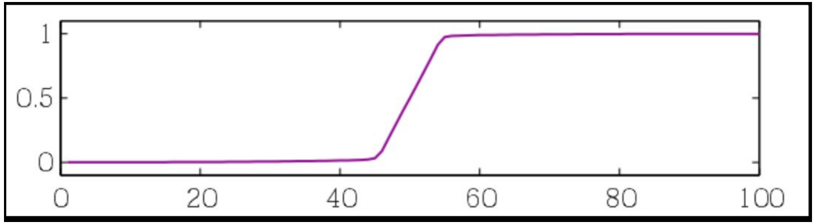




$$J(p) = \frac{1}{K_p} \sum_{q \in R(p)} \exp\left(-\frac{|p-q|^2}{2\sigma_s^2}\right) \cdot \boxed{\exp\left(-\frac{|G(p)-G(q)|^2}{2\sigma_r^2}\right)} \cdot I(q)$$

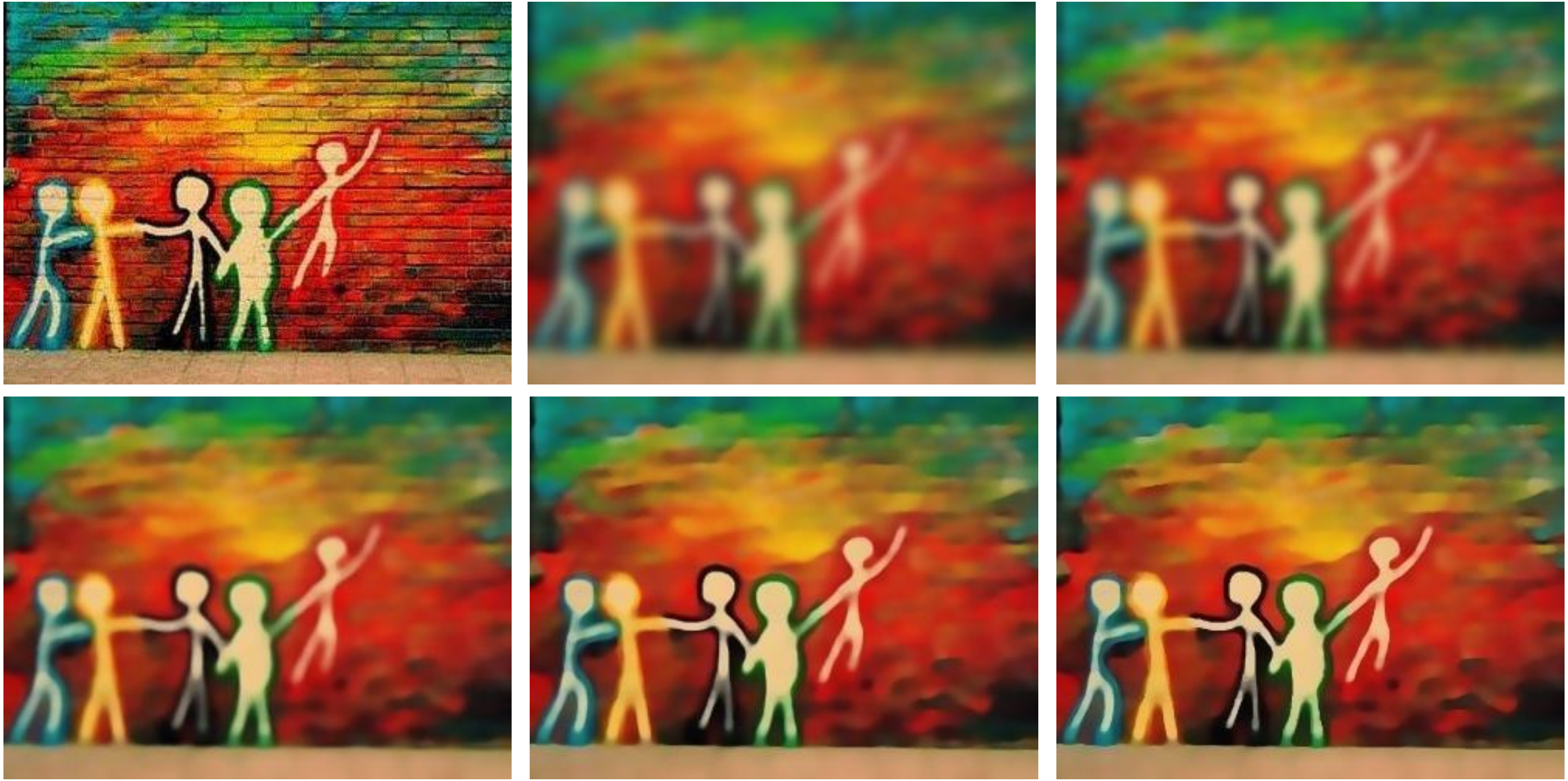
$\neq 1$

The intensity response of a true edge would be a ramp-like structure and the gaussian filter would give an output as shown. To retrieve back the sharpness of the edge we need the intensity variation to come back to original. Here G(p) - G(q) will not be 0. Hence its contribution in the joint bilateral filter will effectively bring back the initial transitional structure of the edge as shown.



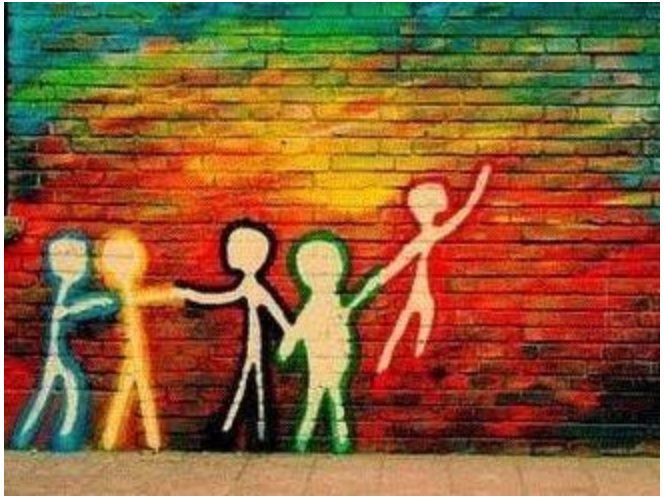
Applying JBF to RGF

1) The output of first iteration would bring the true edge variations closer to the input image variations.
2) In the next iteration if we use the output of first iteration as the guide image for the second iteration the preservation would be sharper as illustrated below.
3) Experimentally we have observed that 4 iterations are enough to preserve true edges and remove noise.
4) Outputs for guide have been shown in the following slides.



a b c a) Input image b) Gaussian of input image i.e. guide for first iteration c) guide for second iteration
d e f d) guide for third iteration e) guide for fourth iteration f) Final Output ($\sigma_s = 3.5$ and $\sigma_r = 25.5$)

Outputs: σ_s = { **0.5 , 1** , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 }

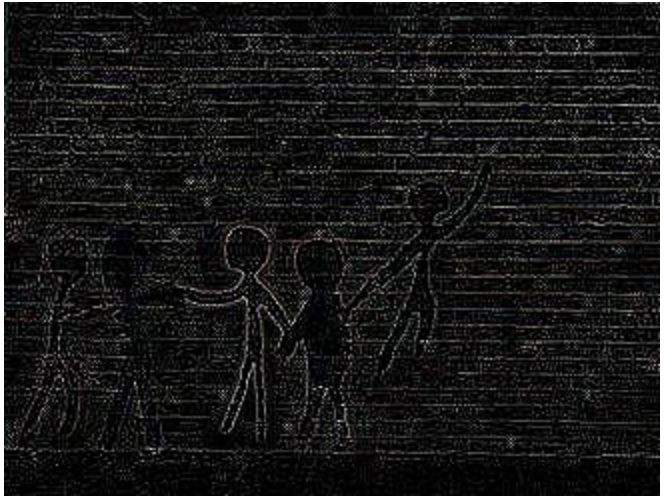


Outputs: σ_s = { 0.5 , 1 , **1.5 , 2** , 2.5 , 3 , 3.5 , 4 }

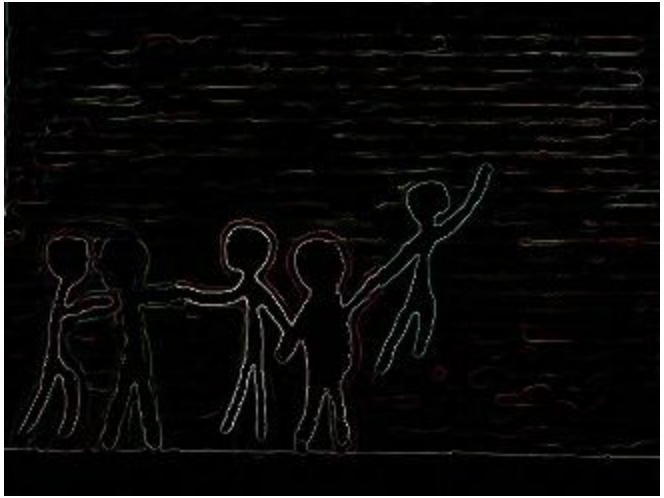
Outputs: σ_s = { 0.5 , 1 , 1.5 , 2 , **2.5 , 3** , 3.5 , 4 }

Outputs: $\sigma_s = \{ 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , \mathbf{3.5} , \mathbf{4} \}$

Laplacian Outputs: σ_s = { **0.5 , 1** , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 }



Laplacian Outputs: $\sigma_s = \{ 0.5 , 1 , \mathbf{1.5} , \mathbf{2} , 2.5 , 3 , 3.5 , 4 \}$



Laplacian Outputs: σ_s = { 0.5 , 1 , 1.5 , 2 , **2.5** , **3** , 3.5 , 4 }

Laplacian Outputs: σ_s = { 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , **3.5 , 4** }

Outputs: σ_s = { **0.5 , 1** , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 }

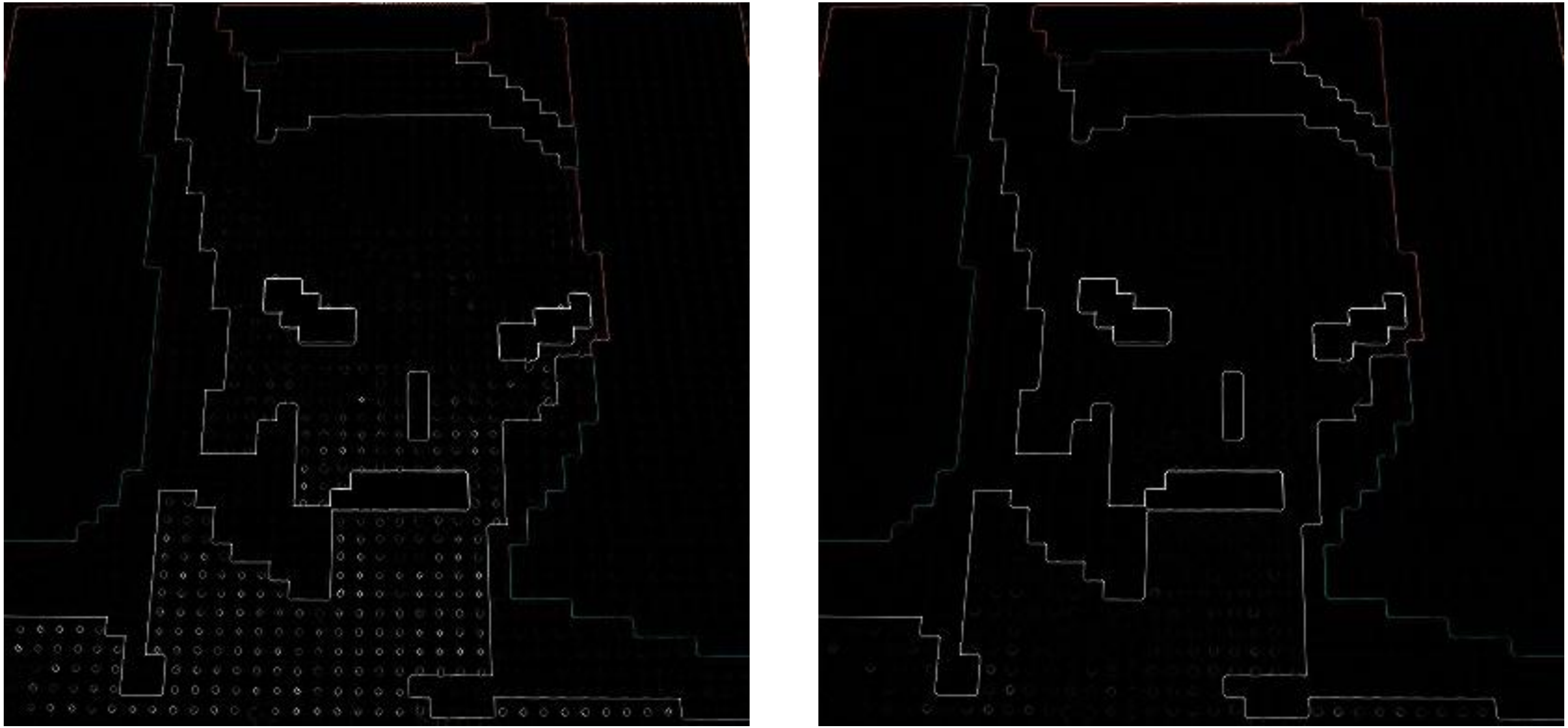
Outputs: σ_s = { 0.5 , 1 , **1.5** , **2** , 2.5 , 3 , 3.5 , 4 }

Outputs: σ_s = { 0.5 , 1 , 1.5 , 2 , **2.5 , 3** , 3.5 , 4 }

Outputs: σ_s = { 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , **3.5 , 4** }

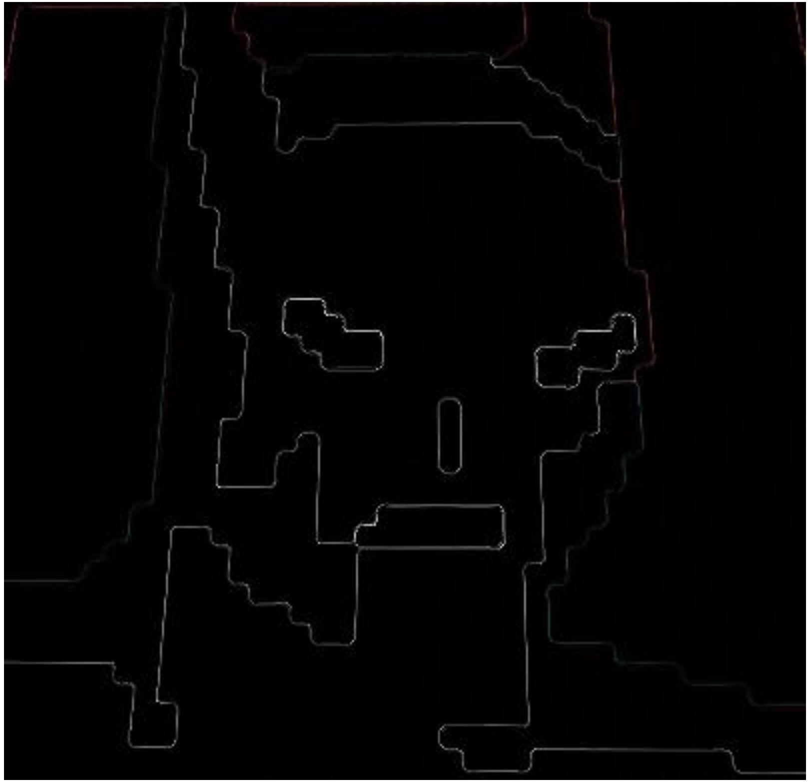
Laplacian Outputs: $\sigma_s = \{ \mathbf{0.5}, \mathbf{1}, 1.5, 2, 2.5, 3, 3.5, 4 \}$

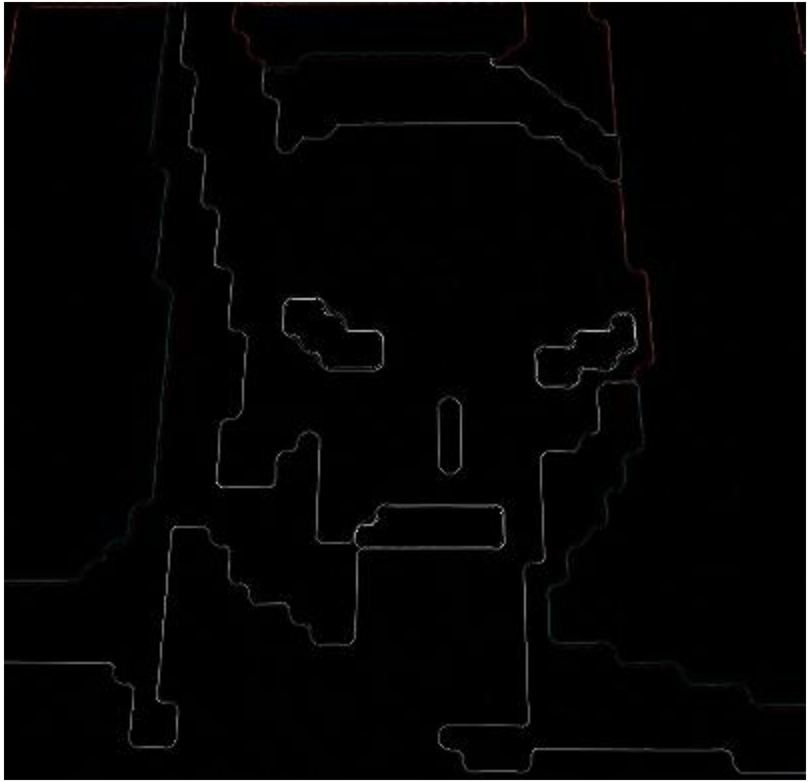
Laplacian Outputs: $\sigma_s = \{ 0.5 , 1 , \mathbf{1.5} , \mathbf{2} , 2.5 , 3 , 3.5 , 4 \}$



Laplacian Outputs: $\sigma_s = \{ 0.5 , 1 , 1.5 , 2 , \mathbf{2.5} , \mathbf{3} , 3.5 , 4 \}$

Laplacian Outputs: $\sigma_s = \{ 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , \mathbf{3.5} , \mathbf{4} \}$





Observations

1) As the value of sigma_s increases the smaller structures in the image are removed and the larger are preserved without disturbing the true edges.
2) The values of sigma_s for the range **{ 0.5 , 1 , 1.5 , 2 , 2.5 , 3 , 3.5 , 4 }** give extremely good transitional observations in the results as can be seen above.
3) The value of sigma_r has been fixed to 25.5 as it shows very good results for most of the input images.

Implementing the Rolling Guidance Filter

- Permutohedral Lattice has been used in the implementation of Rolling Guidance Filter to make the algorithm computationally faster using High dimensional Gaussian filtering.
- High-dimensional filtering is implemented by resampling the input data as points in a high-dimensional space (**splatting**), performing a high-dimensional Gaussian blur on the samples (**blurring**), and then resampling back into the input space (**slicing**).associates an arbitrary position p_i with each value v_i to be filtered, and then mixes values with other values that have nearby positions.

$$\vec{v}_i' = \sum_{j=1}^{n} e^{-\frac{1}{2}|\vec{p}_i - \vec{p}_j|^2} \vec{v}_j$$

Implementing the Rolling Guidance Filter

- The Joint Bilateral Filter is basically a 5-Dimensional Gaussian filter with

$$\vec{p}_i = [\frac{x_i}{\sigma_s}, \frac{y_i}{\sigma_s}, \frac{r_i}{\sigma_c}, \frac{g_i}{\sigma_c}, \frac{b_i}{\sigma_c}]$$

  where corresponding to the pixel i, x_i & y_i are the spatial coordinates; r_i, g_i, b_i are the colour intensities; σ_s refers to the spatial standard deviation of the filter and σ_c is the color-space standard deviation; also,

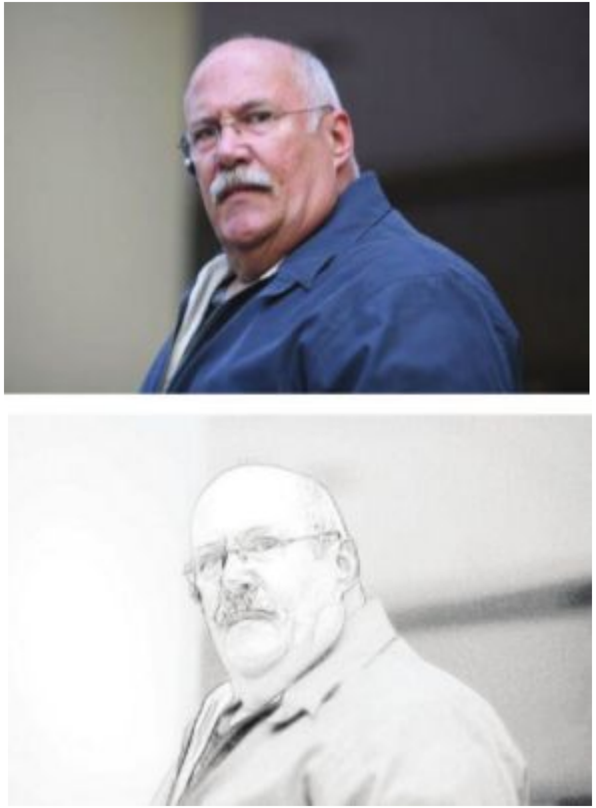
$$\vec{v}_i = [r_i, g_i, b_i, 1]$$

- By deriving p_i from one image (**guide**) and v_i from another (**input**), one can smooth an image in a way that does not cross the edges of another.

Non Photorealistic Rendering (NPR)

- **NPR** stands for **Non-Photorealistic rendering** i.e. enabling an expressive style on an image like digital art. It is a process by which we try to create a diverse styles of television. Pictures and videos.
  The styles include:
  - Painting
  - Drawing
  - Technical illustration
  - Animated cartoons.

Examples of NPR



Pencil Sketch

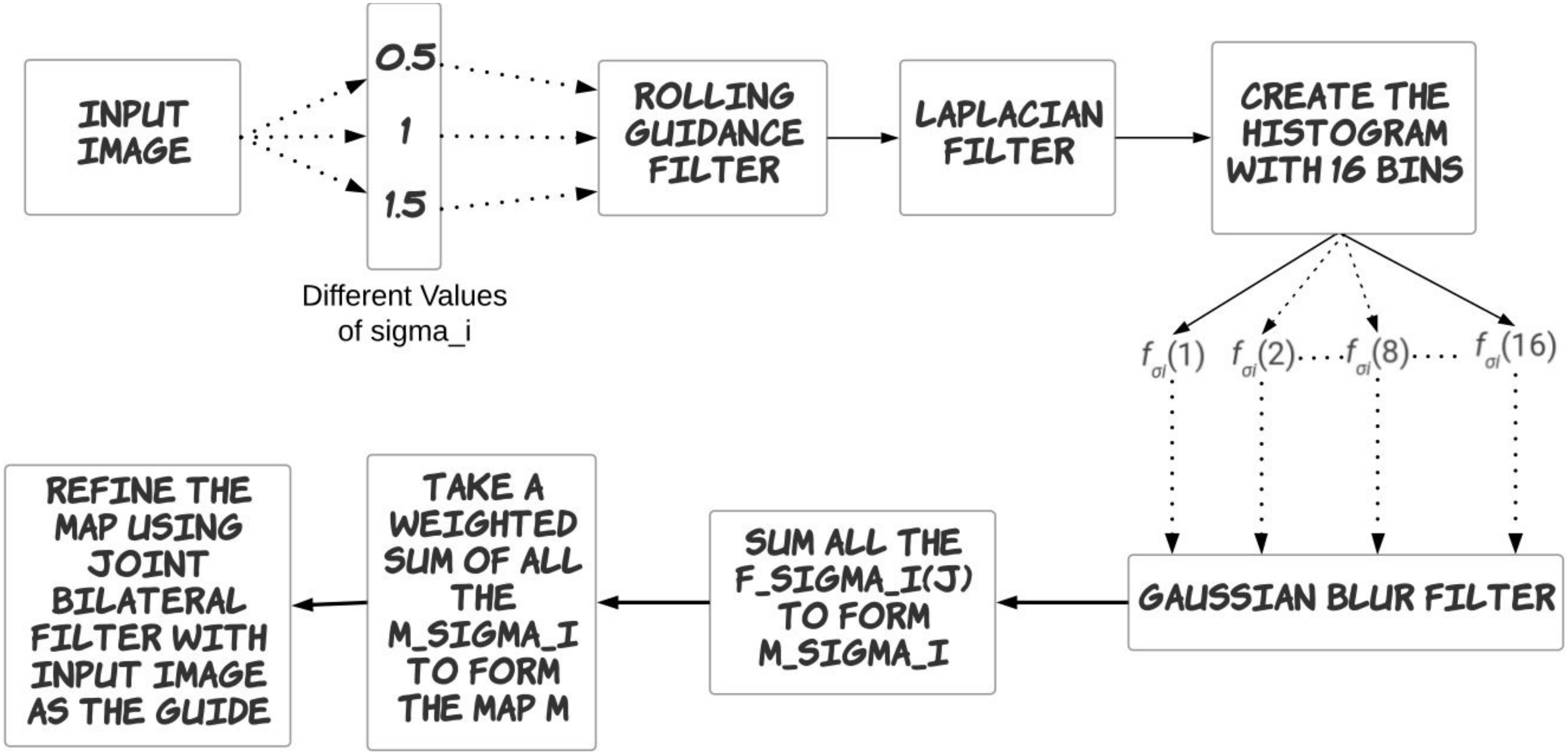
Color Pencil Sketch

Cartoonization

Using the SaSGM Model for NPR

1. NPR techniques can generally be categorized as filtering-based operations with certain number of pre-processing and post-processing methods.
2. The key lies in effectively partitioning textures from dominant edges during the filtering process. For most natural scenes, strong edges usually contain important information and are desired to be preserved in the filtering process.
3. SaSGM Model helps to leverage magnitude and scale information in building spatial guidance for NPR related image processing tasks.
4. We have introduced the SaSGM Model for two NPR techniques i.e. Pencil Sketch and Cartoonization.

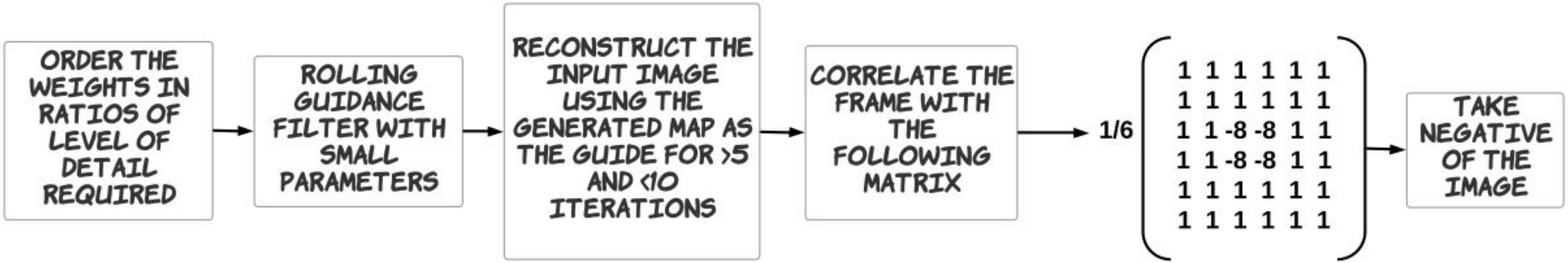
NPR using SaSGM Model



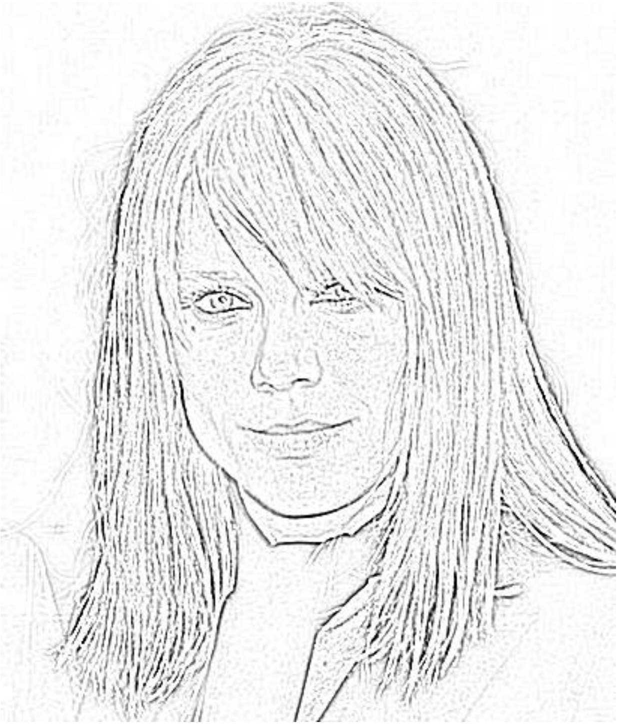
BASIC SASGM MODEL

Observations

1. We tried the code on several different images but the maps obtained were not up to the mark.
2. Following our experiences with the implementation of Rolling Guidance Filter we have come to the conclusion that merely following the equations written in the paper does not yield the same output as that shown in the paper. There are a huge number of pre-processing and post-processing that are not mentioned in our paper and not even referenced.
3. Therefore with our own experimentations and observations, we have modified the output using some post-processing techniques and achieved considerably good results.
4. The following flow chart is the post-processing for generating pencil sketch.



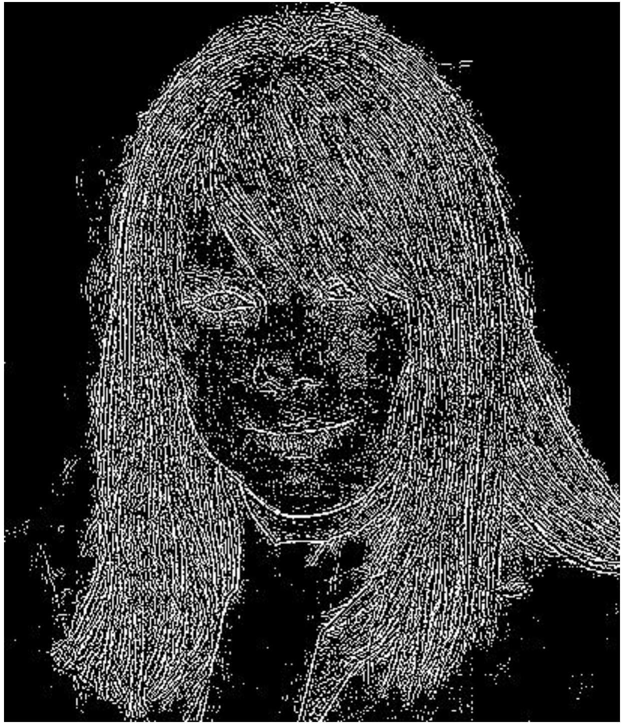
Pencil Sketch of input Image



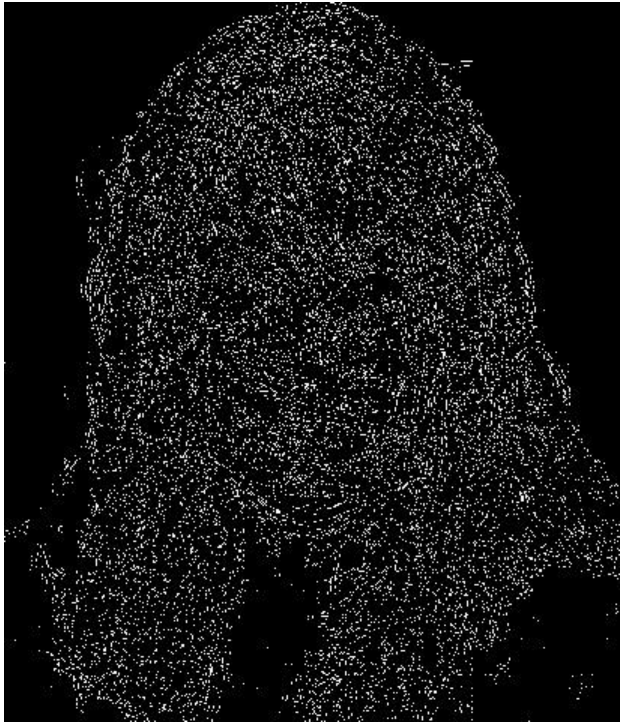
A normal pencil sketch does not remove the small scale features that are basically the texture or noise. They are undesirable and hence should be removed.

Outputs of the Spatial Indicator Function (f) at different j

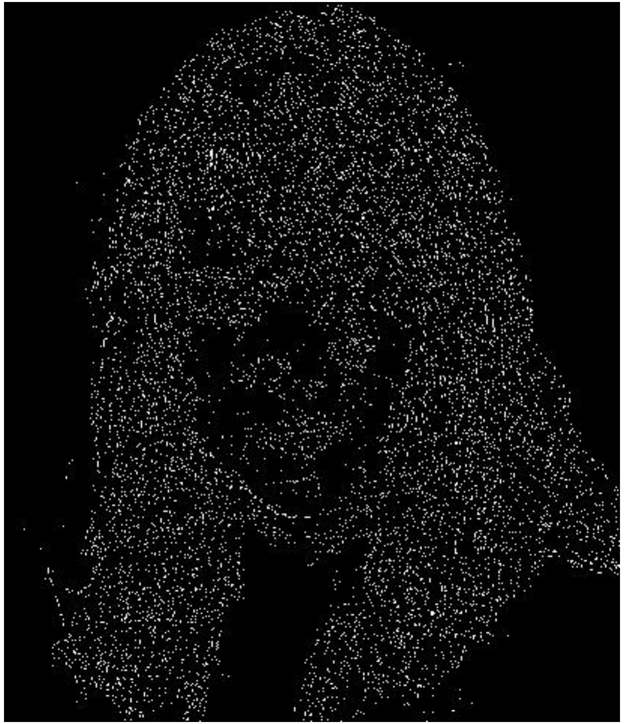
j = 0



j=1

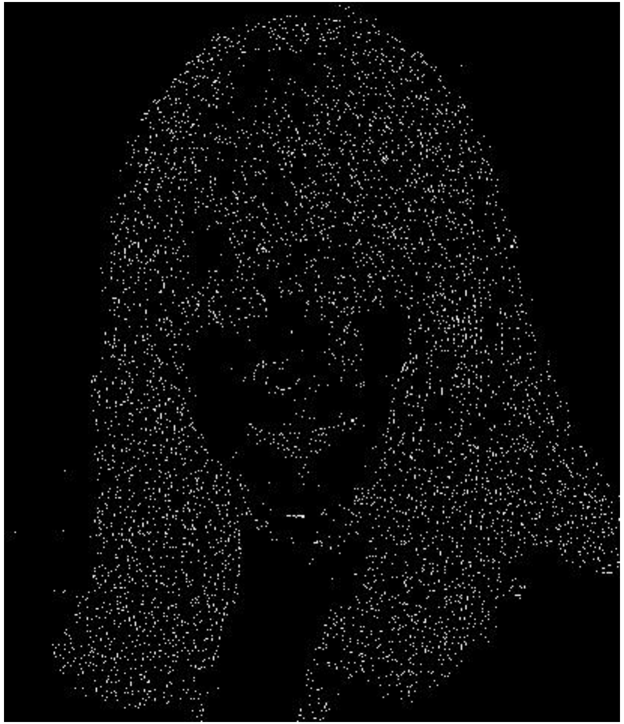


j = 2

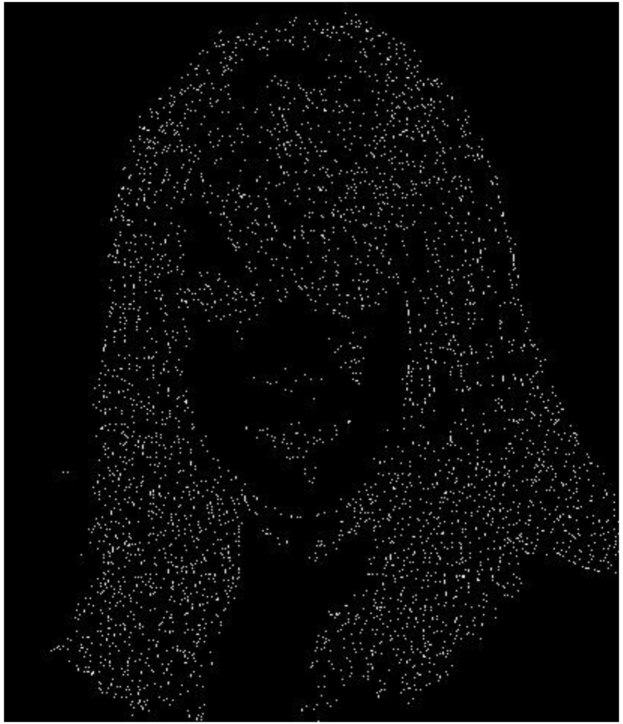


Outputs of the Spatial Indicator Function (f) at different j

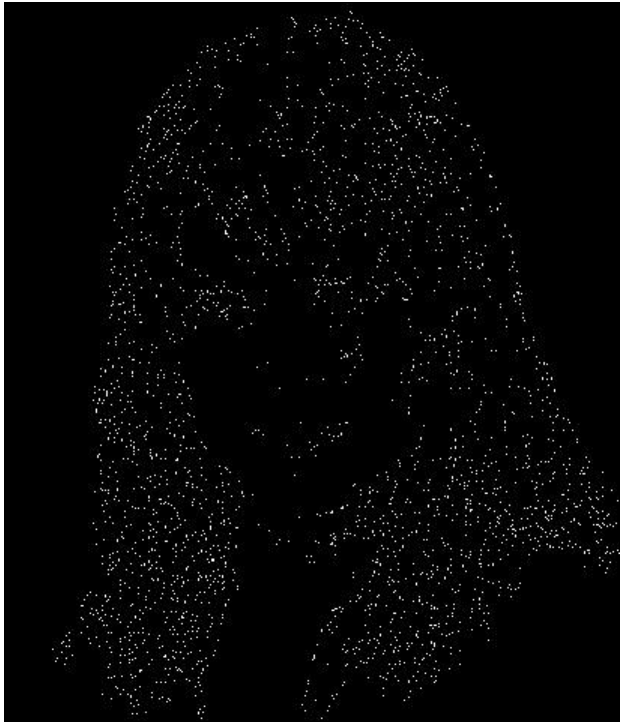
j = 3



j=4

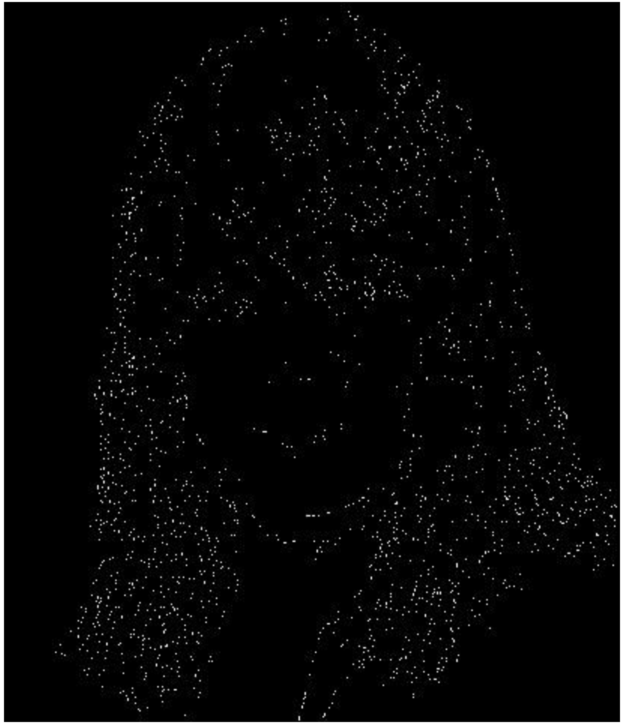


j=5

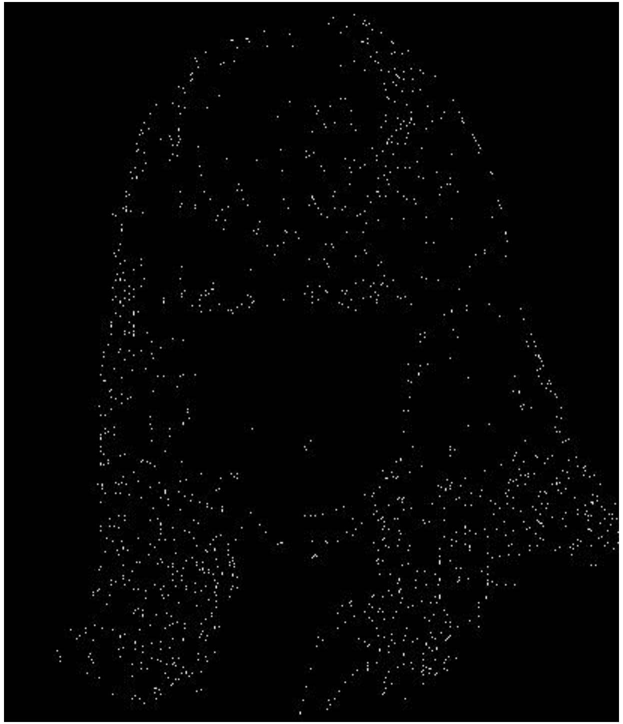


Outputs of the Spatial Indicator Function (f) at different j

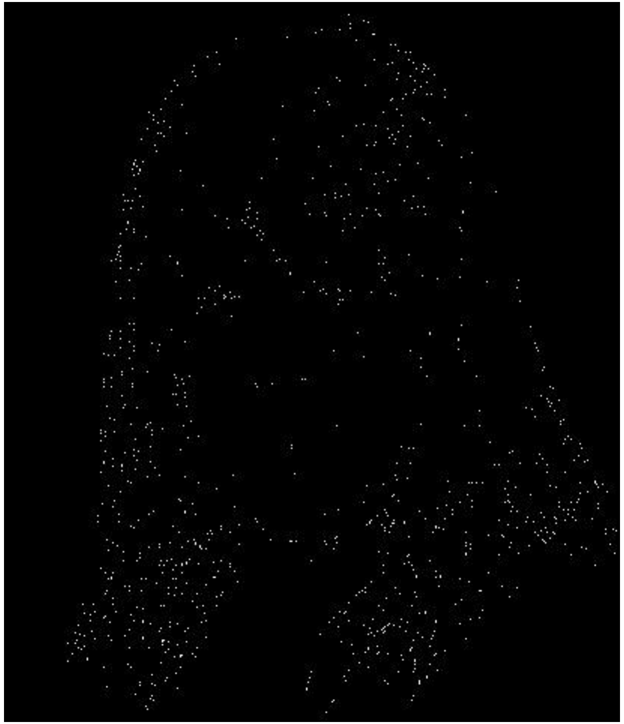
j = 6



j=7

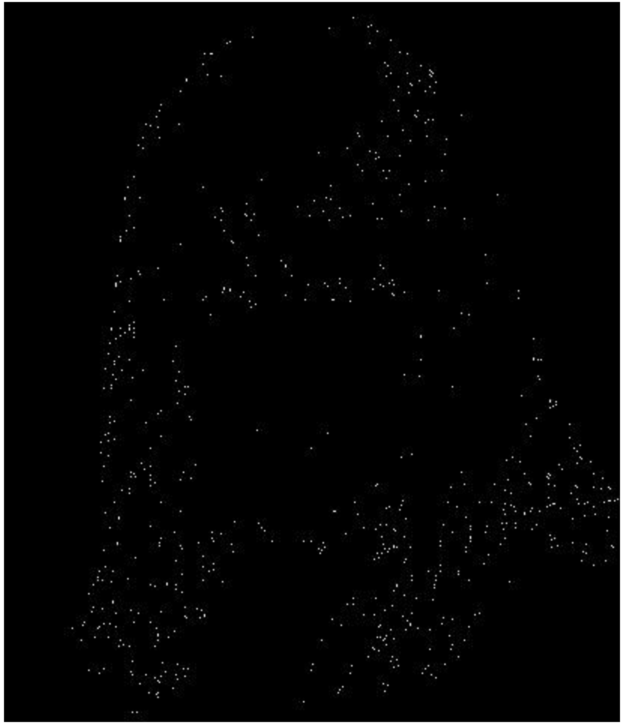


j=8

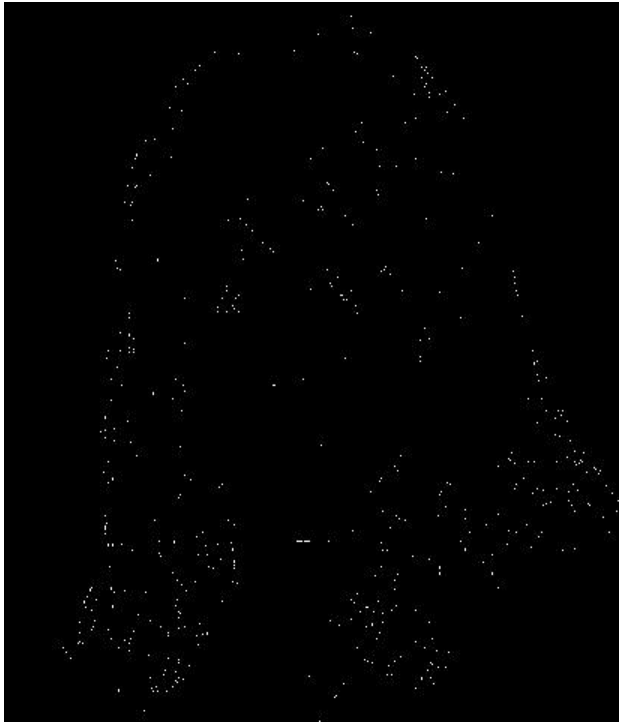


Outputs of the Spatial Indicator Function (f) at different j

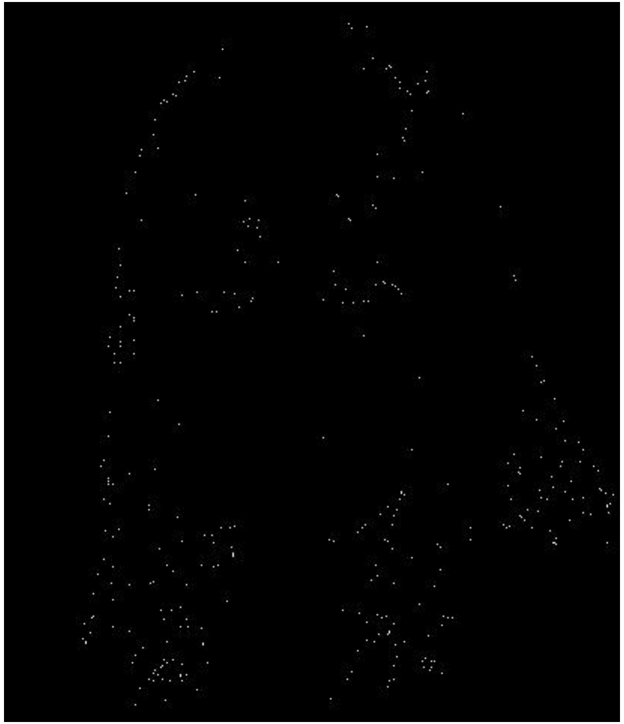
j = 9



j=10

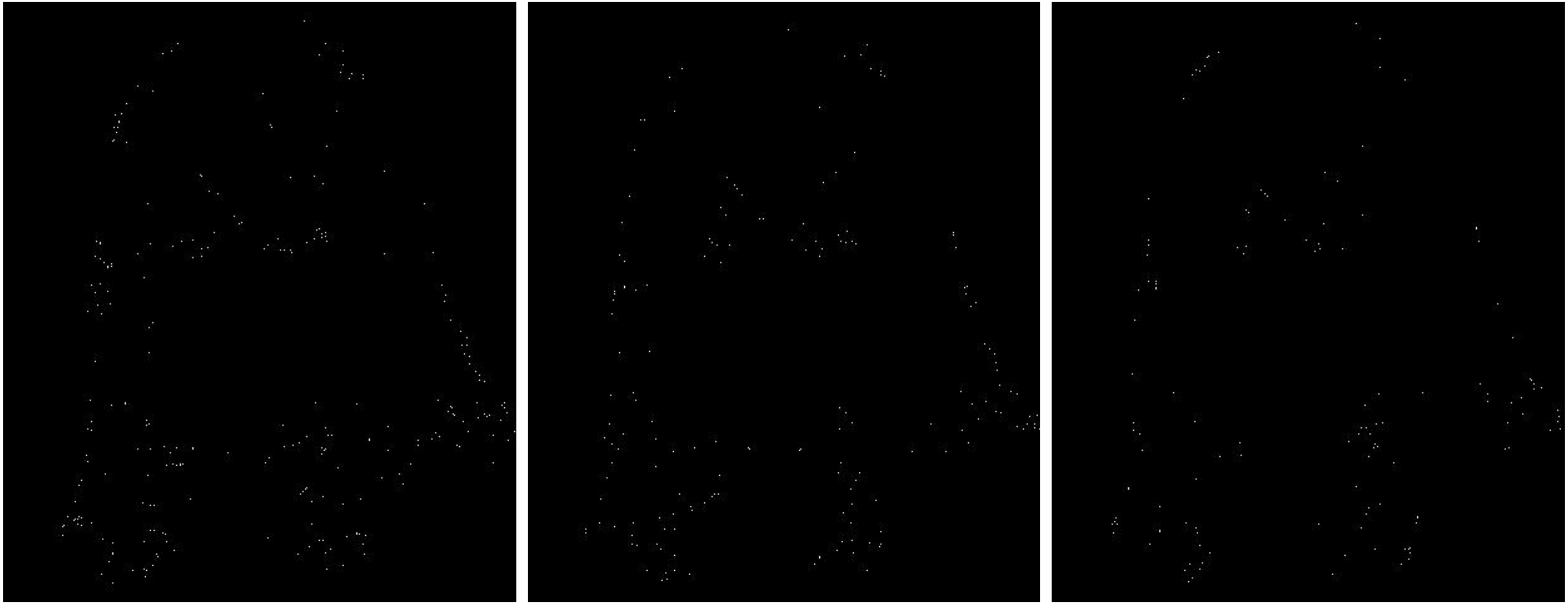


j=11



Outputs of the Spatial Indicator Function (f) at different j

j = 12 j=13 j=14



Outputs of the Spatial Indicator Function (f) at different j

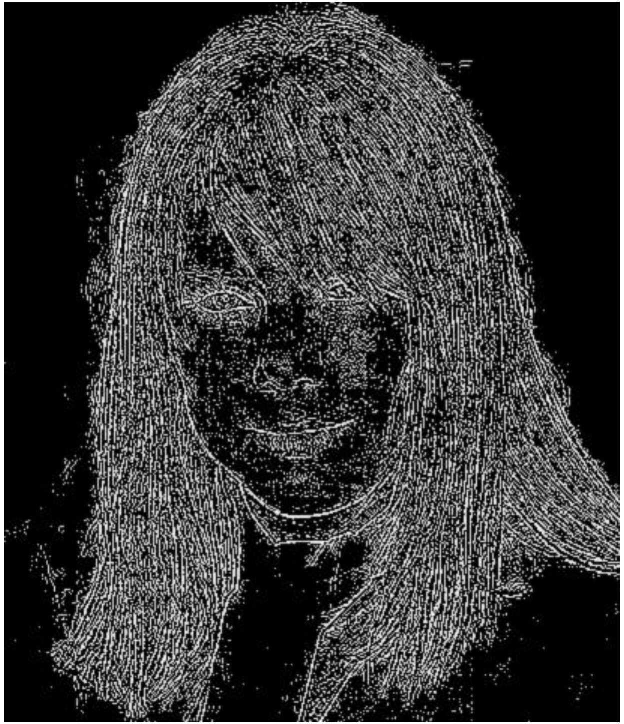
j = 15

Note the decreasing number of non-zero pixels with the increasing value of edge-strengths, i.e. j.

We wish to remove the 'texture', i.e. the pixels corresponding to the minimum strength (j=0). So, we take {1-h(j)} as the weighting coefficient for each of the $f_{\sigma_i}(j)$ which gives minimum weightage to the smallest-strength gradient pixels.

Edge-aware scale maps

$\sigma_i = 0.5$

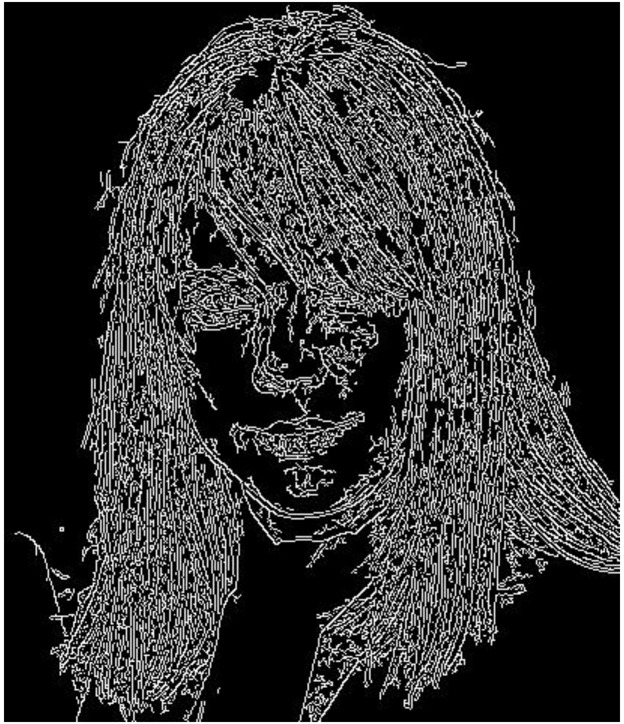


$\sigma_i = 1.0$

$\sigma_i = 1.5$

Edge Representation

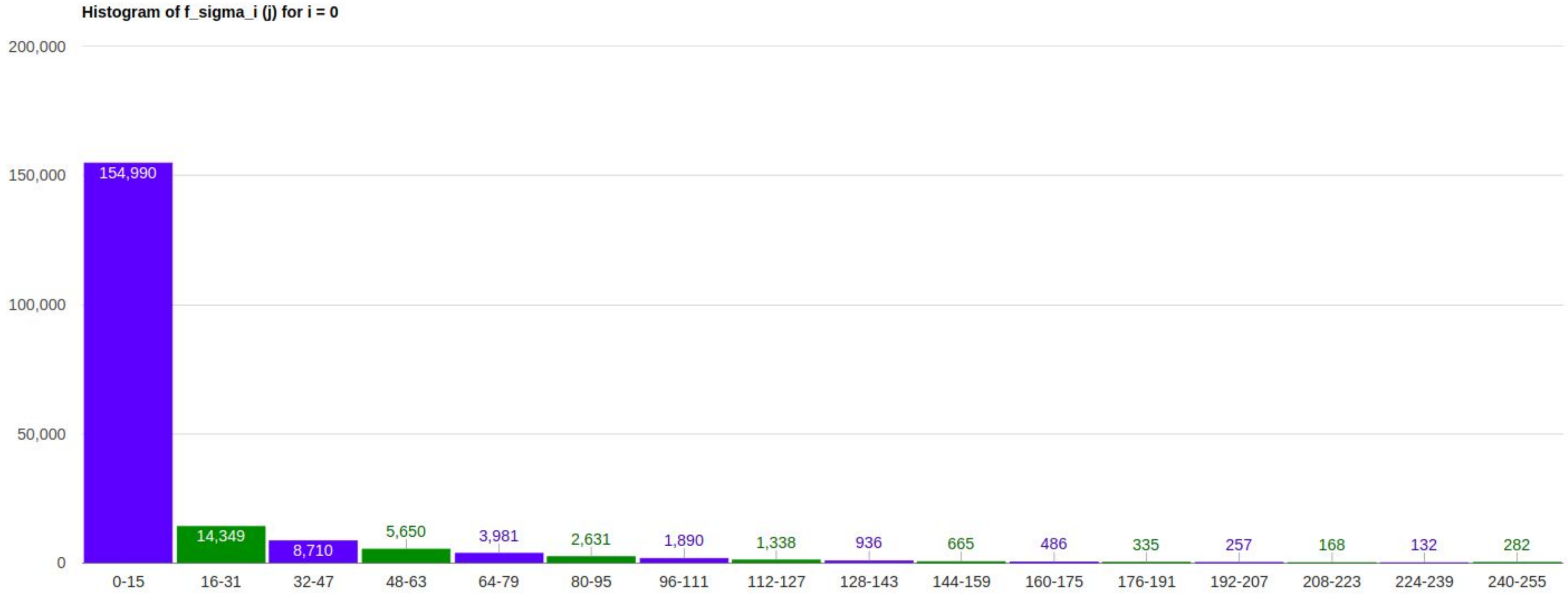
Canny on Input Image



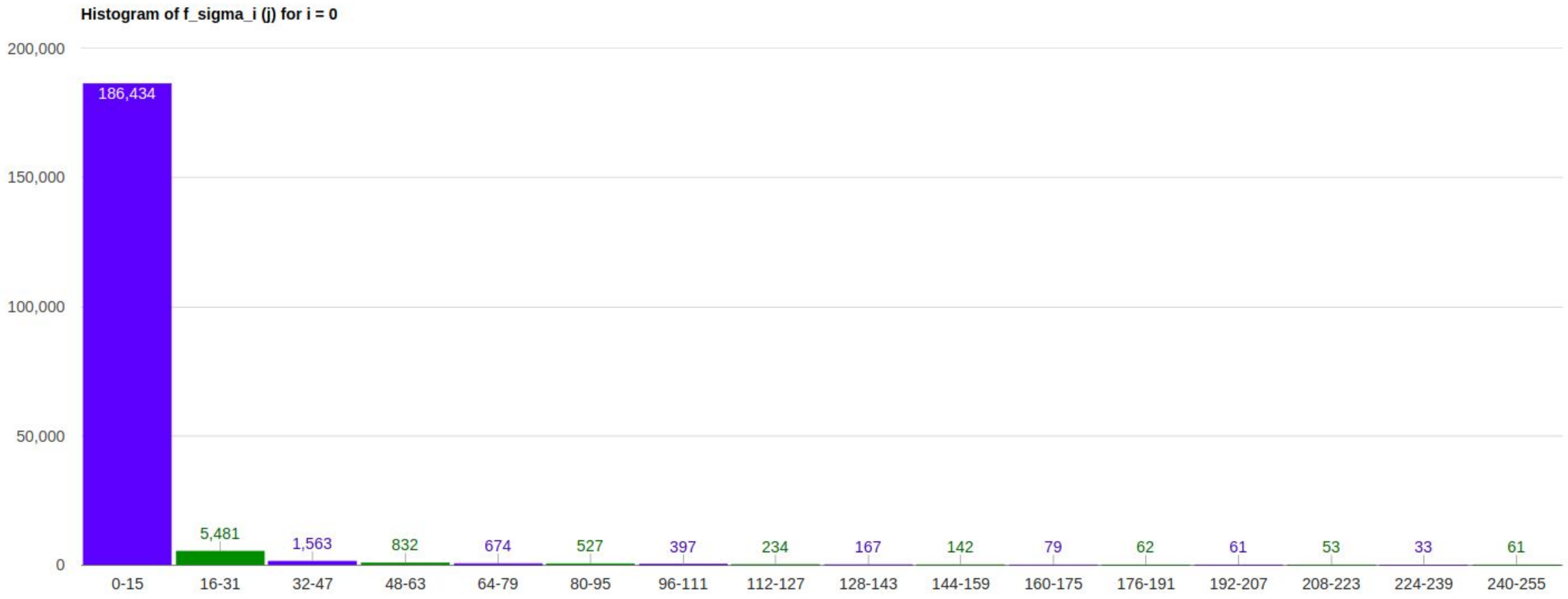
Laplacian on Input Image

SaSGM on input image

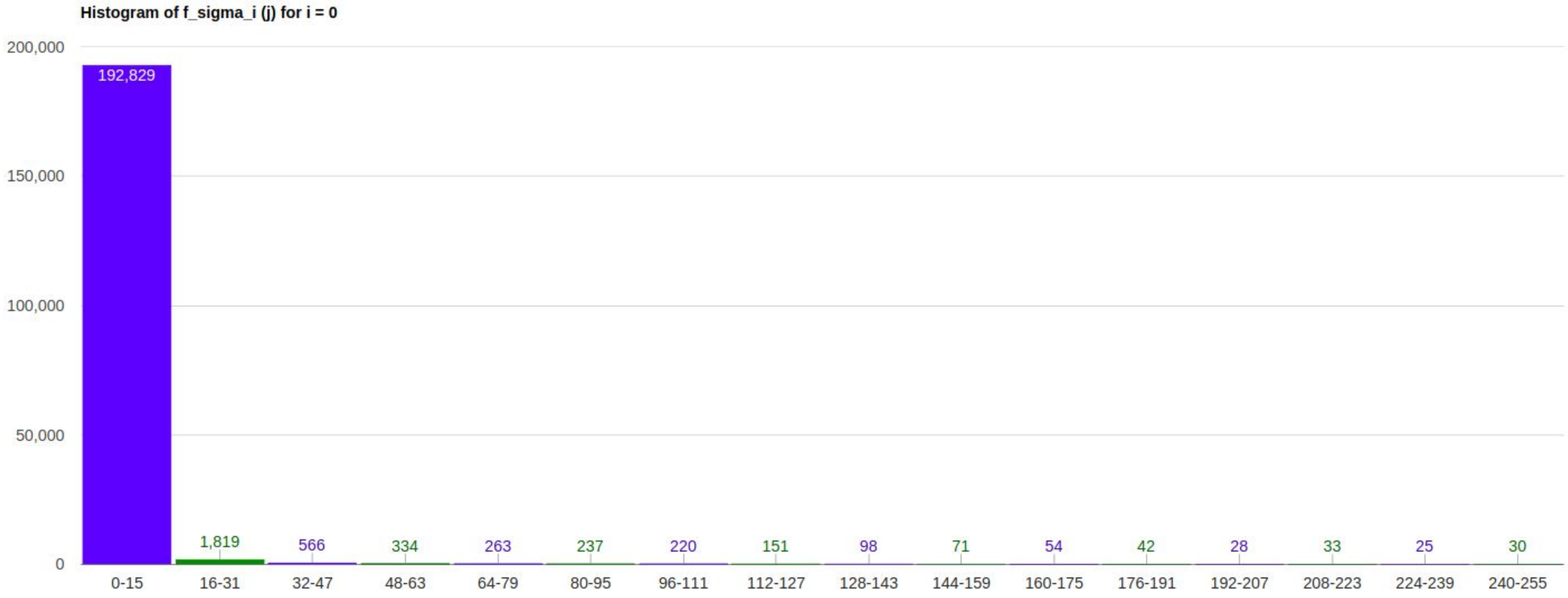
Histogram of i=0 for 16 bins in f_{σ_i}



Histogram of i=1 for 16 bins in f_{σ_i}



Histogram of i=2 for 16 bins in f_{σ_i}



Pencil Sketch

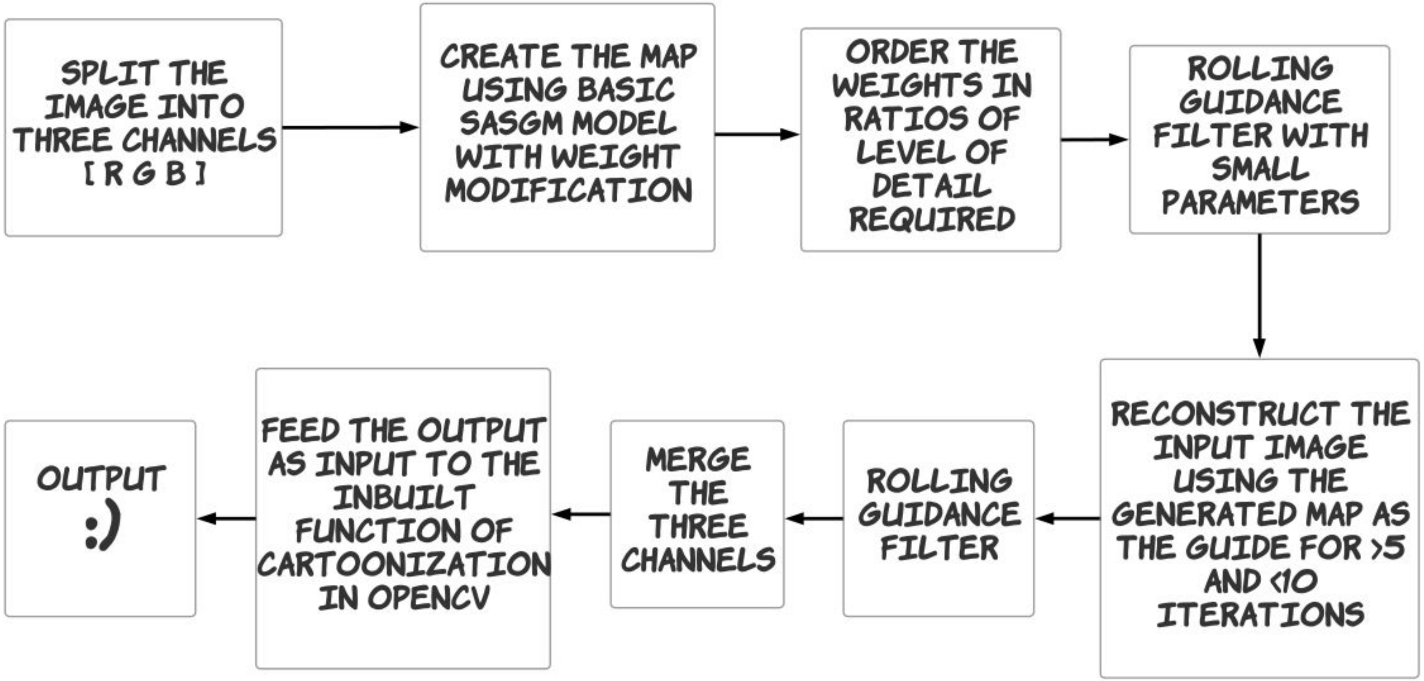
Reconstructed Image

Pencil Sketch (SaSGM)

Our SaSGM model has successfully removed the undesirable blemishes and small scale structures from the woman's face.

Cartoonization using SaSGM Model

Flow chart of cartoonization using the basic SaSGM model at the second step and post-processing after that.



Input Image for cartoonization

Outputs of cartoonization using OpenCV inbuilt function

Note the unwanted black patches all over the image caused by the little impurities and small scale textures present in the image

SaSGM Map for the input image

Outputs of cartoonization after preprocessing with our SaSGM

All the unwanted responses have been removed by the pre processing of the image using SaSGM and we obtain a better output than the before.

Parameter Selection Constraint

Parameter	Values	Explanation
σ_i	{ 0.5 , 1 , 1.5 }	For values greater than 1.5 the scale responses becomes zero when laplacian filter is applied. Hence their effective contribution in output is nullified. For values less than 0.5 the scale response captures mainly noise which is undesirable for NPR.
nb	16	Less than 16 bins doesn't effectively divide the gradient strengths, while greater than 16 bins just increases the computations and hence, reduces the efficiency.

Map coefficients (M_{σ_i})	If (i==0):
	1/16
	else:
	(i+1)/8
		i=0 represents the smallest level of details in the input image, hence its contribution to the output Map is minimum. If the value is taken to be 0 then the reconstructed image loses its sharpness. For i>0, the coefficients increase with increase in the scale value/level of the edges.

Values of image sizes and number of iterations of RGF	Size:500x500 Iterations: <10	These constraints are due to limited memory available to the function call stack. The execution will abort if memory usage exceeds the limit.

Our Learning

1) Understanding of concept of Rolling Guidance Filter.
2) Using RGF and Joint Bilateral filter for various pre-processing and post-processing operations.
3) Analysis and implementation of SaSGM model.
4) Applying our model to two types of NPR techniques.

Thank you.